In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

3/17/65
b

OPTIMAL VALUE ESTIMATION APPLIED TO THE REDISTRICTING PROBLEM

A THESIS

Presented to

The Faculty of the Graduate Division

by

William Walter Swart

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

Georgia Institute of Technology

December, 1967

OPTIMAL VALUE ESTIMATION APPLIED

TO THE REDISTRICTING PROBLEM

Approved:

Chairman

Date approved by Chairman: April 29-1968

ACKNOWLEDGMENTS

I wish to express my gratitude to Dr. D. E. Fyffe for his excellent advice and encouragement during the course of this study. In addition, I wish to thank Dr. W. W. Hines for his careful and constructive critique of this work, and Dr. K. Chu for serving on my Thesis Committee.

I also wish to express my appreciation to CROND, Inc. and, in particular, to Messrs. Jack Whelan, Harry Siegfeldt, Paul Zitlau, Jim Weaver, and Sidney Hess for their support of this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

This thesis develops a procedure that may be used to estimate the objective function value of the optimal solution to an optimization problem. The only requirement for the procedure to be applicable is that it be practical to obtain feasible solutions to the optimization problem.

The approach consists of investigating the distribution of feasible solution to the problem and using statistical procedures to estimate a bound on the range of feasible solutions. This bound is then taken as an estimate of the optimal solution to the problem.

The procedure is developed for and applied to the redistricting problem as formulated by Weaver and Hess (4). A generalized algorithm is presented in Appendix I.

One item of particular interest in this thesis is the use of a heretofore largely neglected method of parameter estimation--that of estimating the parameters of the population distribution which characterize a set of data by minimizing the test statistic of the chi-square goodness-of-fit test.

Listings of the computer programs used for this research are included in Appendix II.

CHAPTER I

INTRODUCTION

General

Decision problems arise in many fields. Typically, such problems require that decision variables satisfy certain constraints which are imposed by the problem situation. Depending on the constraint set, there may be several feasible choices for the decision variables. As a result, the decision-maker must seek the best set of decision variables which satisfy the constraints. The term "best" is relative and implies that the decision-maker has a criterion by which to determine the "goodness" of a solution to the problem.

If the criterion and the constraints can be expressed in analytical relationships between the decision variables, then the problem may be formulated as follows:

$$\begin{aligned} &\underset{\text{over all } x}{\text{Maximize (Minimize)}} && f(X) \\ &\text{Subject to} && g_j(X) \geq 0 \end{aligned} \tag{1.1}$$

Where:

- X is an N-dimensional vector of decision variables.
- $f(X)$ is a criterion function expressing a measure of effectiveness of the solution, and
- $g_j(X)$ is a M-dimensional constraint set imposed on the problem.

Problems such as (1.1) belong to the general class of constrained optimization problems. Because these problems occur frequently, a great deal of work has been done on methods for their solution. Unfortunately, however, efficient and practical techniques are available only for problems which can be analytically structured to meet some rather stringent conditions. A review and discussion of these may be found in Reference 1.

In realistic situations, the decision-maker must find a solution to a problem even though the analytical formulation does not satisfy the necessary conditions for solution by one of the available techniques. Under such circumstances, it is generally more practical to find a solution that satisfies the constraints of the problem (a feasible solution) than to seek the solution which optimizes the criterion *and* satisfies the constraints simultaneously (the optimal solution).

Whenever it is not practical to find the optimal solution to a problem and the decision-maker must be satisfied with finding a feasible solution to the problem, the question then arises as to how good this feasible solution is with respect to the optimal. In many situations, an answer which is arbitrarily close to the optimum may be accepted as satisfactory. However, before a decision can be made as to when a feasible solution is arbitrarily close to the optimum, the optimum itself must be known or estimated.

Objective and Purpose

The objective of this thesis is to develop a procedure whereby a lower bound on the value of the criterion function of an optimization problem may be found. The purpose is to provide a standard against which feasible solutions to the problem may be evaluated in terms of their proximity to the estimated lower bound.

Background to the Problem

The research was motivated by a situation encountered by the author during his association with a group of individuals interested in the application of quantitative techniques to the problem of non-partisan redistricting. This group developed an integer programming model as a statement of the redistricting problem. In essence, the model represented a location-allocation problem, the solution of which would give the optimal location of Legislative District centers and allocation of population units to these centers.

The measure of effectiveness of the integer programming model is the sum of the squared distances from each population unit center to the center of the Legislative District center to which it is assigned times the population of the particular population unit. This measure of effectiveness was used because, by minimizing it, the resulting geometrical shape of the Legislative Districts would tend to be compact. The constraints to the integer programming formulation specified the number of Legislative Districts to be formed and required that the Legislative Districts have approximately equal populations.

The integer programming formulation appeared to be a very good model of the redistricting problem. Unfortunately, the dimensionality of the problem was such that it was impractical to find the solution. As a result, efforts were directed toward solving an earlier formulation of the problem. This formulation differs from the original in that single voters are allocated to a set of specified Legislative District centers. This earlier formulation of the problem falls into the category of allocation problems and can be solved by well-known linear programming algorithms.

The measure of effectiveness of the allocation model is the sum of the squared distances from the center of each population unit times the number of voters of that particular population unit assigned to the Legislative District. The constraint set to this model specifies that the resulting Legislative Districts have population equality.

At this point, it is worthwhile to note that the solution to the allocation problem satisfies the constraints to the location-allocation problem. Therefore, the solutions obtained from the allocation problem are feasible solutions to the location-allocation problem. The problem which remains is that of finding the "best" solution to the location-allocation problem from these feasible solutions.

Motivation for Research

The motivation to work on this particular topic stemmed from the apparent lack of published work on estimating, by some rational means, the optimal objective function value to an optimization problem given that it was not practical to solve it.

The particular application which was considered was the estimation of the objective function value to the optimal solution of the location-allocation problem. This was relevant because redistricting plans obtained by the allocation problem could not be compared to any standard in order to determine their "goodness" with respect to the best redistricting plan that could be obtained. As a consequence, the procedure followed at that time to obtain redistricting plans was to solve the allocation problem with different sets of Legislative District centers until a particular plan was selected by some arbitrary criteria of "goodness".

With an estimate of the objective function value to the optimal solution of the location-allocation problem, an absolute standard of "goodness" for the solution of the allocation problem would be available, eliminating the need of evaluating these solutions with respect to arbitrary criteria.

Approach

The set of feasible solutions to the location-allocation problem is a subset of the set of solutions to the allocation problem. It was decided, therefore, to investigate the distribution of solutions to the allocation problem and to use statistical procedures to estimate a bound on the optimal solution to the location-allocation problem. This was accomplished by taking sets of random Legislative District locations and solving the allocation problem for each.

The results obtained were classified and a histogram constructed. From the shape of the histogram, it was hypothesized that a Weibull distribution might characterize the population distribution of feasible solutions. Before a statistical test was made on this hypothesis, the parameters of the particular Weibull distribution characterizing the population distribution of feasible solutions were estimated by minimizing the chi-square test statistic for goodness-of-fit.

With the distribution parameters so estimated, the chi-square goodness-of-fit test was used to test the hypothesis that the population distribution was as specified. If the test resulted in failure to reject the hypothesis, then the assumption was made that, in fact, the population distribution was Weibull.

An estimate of the lower bound on the objective function of the allocation problem was obtained by minimizing the location parameter, denoted by γ, subject to the constraint that the original conclusion that the population distribution was Weibull was not rejected. This value of the location parameter, denoted by γ^* was taken to be a lower bound to the objective function values of the location-allocation problem.

CHAPTER II

LITERATURE SURVEY

The Weaver and Hess Algorithm

The specific problem considered in this thesis was originally stated by Weaver and Hess. This chapter is a development of their published work (4).

Redistricting deals with the problem of establishing boundary lines in a geographical area for the purpose of dividing the area into a set of Legislative Districts, or L.D.'s. Each district is then entitled to political representation in the local government.

In 1962 the U. S. Supreme Court ruled that federal courts must decide constitutional challenges to state legislative representation at the suit of qualified voters who claimed that present reapportionment did not offer them equal opportunities for representation (6). As a result of this ruling, a series of court cases, known as the reapportionment cases (4), resulted in the following requirements to be established for legislative districts:

1. Each legislative district should have substantial population equality.
2. All legislative districts should be contiguous.

Even though no specific meaning was given to the terms "substantial population equality" and "contiguous," they have been interpreted as follows: Substantial population equality implies the average population per district plus or minus some fraction of this amount. Contiguous means that each district should be a single parcel of land bordering with at least one other district.

Any geographical area may be subdivided into a set of smaller areas, each having associated with it a certain population. Weaver and Hess approached the problem of redistricting by taking such a subdivision of the area to be redistricted and denoting each subdivision by a point. Each point had associated with it the coordinates of the center of the subdivision and its corresponding population. Legislative Districts were then composed by selecting sets of points so that each set represented one Legislative District. The actual subdivision used by Weaver and Hess were Enumeration Districts, henceforth referred to as E.D.'s. An E.D. is a population unit used by the U. S. Census Bureau.

To determine a criterion by which to select the E.D.'s that were to compose each L.D., the main consideration was given to the requirement of contiguity. To facilitate later work, the concept of Compact Legislative Districts was used to express contiguity. Compactness is meant to imply that the geometric shape of the resulting Legislative District is regular. (For example, the most regular or compact geometric shape is a circle.) Therefore, it was conjectured that a compact Legislative District would also be contiguous. To obtain a quantitative measure of compactness, Weaver and Hess used the relation $d_{ij}^2 P_j$. The d_{ij} represents the distance between the point representing the jth E.D. and the population center of the ith L.D. P_j represents the population of the jth E.D.

If the assignment of E.D.'s to L.D.'s is made so as to minimize the measure of compactness, the resulting assignment will tend to cluster the E.D.'s assigned to a particular L.D. around the population center of that L.D. The resulting L.D. can then be expected to be compact, although this will not always be the case.

The location of the population centers of the L.D.'s were determined by considering each point representing an E.D. as a possible candidate. The points finally chosen were those that minimized the total measure of compactness.

To account for the requirement of substantial population equality between legislative districts, a constraint was placed on the minimization of the total measure of compactness stating that the population of each district should be plus or minus some fraction of the average overall legislative districts.

To summarize what has been said thus far, Weaver and Hess formulated the redistricting problem so that the solution to it would yield the best possible location of legislative district centers and allocation of enumeration districts to these centers such that the total measure of compactness was minimized. Expressed quantitatively, this formulation is:

$$\text{Minimize } f(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2 \, P_j x_{ij}$$

Subject to:

$$\sum_{j=1}^{n} P_j x_{ij} \leqq (P_j)/k(1+a)x_{ii} \qquad i = 1,n$$

$$\sum_{j=1}^{n} P_j x_{ij} \geqq (P_j)/k(1-a)x_{ii} \qquad i = 1,n$$

$$\sum_{i=1}^{n} x_{ii} = k$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,n$$

$$x_{ij} = x_{ij}^2 \Rightarrow x_{ij} = \{{}^0_1 \qquad i = 1,n; \quad j = 1,n$$

Where:

i - index of the ith L.D.

j - index of the jth E.D.

k - number of L.D.'s to be formed.

a - allowable population deviation between L.D.'s expressed as a fraction of the average population between L.D.'s.

n - number of E.D.'s composing area to be redistricted.

d_{ij} - linear distance from the jth E.D. to the population center of the ith L.D.

x_{ij} - zero-one decision variable. It takes on a value of one if the jth E.D. is assigned to the ith L.D., and zero otherwise.

The first two constraint sets indicate that the sum of the populations of the E.D.'s assigned to the ith L.D. should be within the allowable population deviation from the average.

The third constraint specifies the number of L.D.'s that are to be formed. The fourth constraint set specifies that each E.D. may only be assigned to one L.D. The fifth constraint set specifies the nature of the decision variables.

Since, as indicated in Chapter I, this problem is computationally impractical to solve, Weaver and Hess developed another formulation of the problem, the solution of which would yield redistricting plans that satisfied the constraint of substantial population equality and that were compact. In other words, a procedure was developed that yielded feasible, but not necessarily optimal, redistricting plans.

The approach taken to develop a formulation of the problem, the solution of which yielded feasible redistricting plans differed from the original problem in two ways. Instead of considering the allocation of the complete population of an E.D. to one L.D., it considers the allocation of individual members of the E.D. population to L.D.'s. Also, instead of determining the optimal location of L.D. population centers together with the optimal allocation of population to these, it must be given a set of L.D. population centers to which it then optimally allocates the population of the area to be redistricted.

The mathematical formulation of this problem is:

$$\text{Minimize } f(Y) = \sum_{i=1}^{k} \sum_{j=1}^{n} d_{ij}^2 \, y_{ij}$$

$$\text{Subject to: } \sum_{j=1}^{n} y_{ij} = (\sum_{k} P_j)/k \qquad i = 1,k$$

$$\sum_{i=1}^{k} y_{ij} = P_j \qquad j = 1,n$$

$$y_{ij} \geqq 0 \qquad i = 1,k;\ j = 1,n$$

where

i - index of population units.

j - index of Legislative Districts.

k - number of Legislative Districts to be formed.

n - number of population units that compose the area to be redistricted.

y_{ij} - number of voters from the ith population unit assigned to the jth Legislative District.

P_j - population of the jth Legislative District.

Note that the format of this formulation corresponds to a transportation problem.

The decision variable of the feasible solution problem is the quantity y_{ij}. This represents the number of voters from the jth E.D.

assigned to the ith L.D. Even though this decision variable may make the total measure of compactness smaller than if the total population of an E.D. had been assigned to one L.D., the consequences that this may bring about are undesirable. This can be explained by the fact that the redistricting of geographical entities is not static. Generally, periodic redistricting of these entities will be necessary to account for the effects of population trends and changes on the "one man one vote" requirement. As a result, new Legislative District boundaries will have to be established and defined after each redistricting. Without a predetermined system, such as defining various Enumeration Districts as being the boundaries of Legislative Districts, the implementation of the redistricting plans may be extremely difficult.

The constraints to the feasible solution problem specify that each Legislative District should have the same population. Since the requirements set up by the Supreme Court are not this stringent, Weaver and Hess formulated a hueristic procedure which attempts to better the solution obtained by the feasible solution problem. The basic idea of the procedure is to take all the E.D.'s that do not have all their population assigned to one L.D. and assign all the population of that E.D. to the L.D. which, in the solution to the feasible solution problem, had the most population of the E.D. assigned to it. Once this was done, the population center of the altered L.D. was recomputed and the feasible solution problem was solved again with the new population centers as data. This process was continued until either a specified number of attempts had been made or the same solution was obtained twice in a row.

In spite of the hueristic procedure, there is no guarantee that the solutions obtained from the feasible solution problem will be optimal. It is possible that by chance the population centers that are given as data to this problem are the same as those that would be given by solution to the original problem. In that case the redistricting plan resulting from the feasible solution problem would be the optimal redistricting plan. Nevertheless, even if this was to occur, there is no way of knowing this. Therefore, it still would be useful to have an estimate of the lower bound on the solution values of the feasible solution problem. This is precisely the purpose of this thesis.

Other

No published work on the subject of optimal value estimation was found. References consulted in conjunction with the various statistical and optimization techniques used are listed in the Bibliography.

CHAPTER III

METHOD OF PROCEDURE

This chapter describes the actual details of the procedure used to estimate a lower bound to the optimal objective function value of the integer-programming formulation of the redistricting problem. The actual data used was obtained from a test problem used by Weaver and Hess (4). The problem was to form five Legislative Districts from a geographical area subdivided into 65 Enumeration Districts.

Obtaining Feasible Solutions

As indicated in Chapter I, a lower bound to the original problem was to be estimated from the population distribution of the results to the feasible solution problem. To estimate this distribution a random sample of 50 solutions to the latter was obtained. This was accomplished by reasoning that if 50 sets of Legislative District population centers selected at random were used as data to this problem, the corresponding solutions would also be random.

To select Legislative District population centers at random, each Enumeration District was identified by an integer number between one and 65. Fifty sets of five different random integers were selected. In each of these 50 sets, the five random integers were taken to represent Enumeration Districts. The geographical location of the population centers of these were used as the locations of the Legislative District population centers.

For each of the 50 sets of random population centers so obtained, the feasible solution problem was solved.

To determine what population distribution could characterize the results listed in Table 1, a histogram of this data, exhibited in Figure 1, was constructed.

Table 1. Results of Feasible Solution Problem for 50 Sets of Legislative District Centers Selected at Random

Class Interval	Class Boundaries	Class Mark	Frequency
1	0.9500 - 1.1640	1.0570	4
2	1.1645 - 1.3780	1.2711	9
3	1.3785 - 1.5920	1.4852	11
4	1.5925 - 1.8060	1.6993	5
5	1.8065 - 2.0200	1.9134	4
6	2.0205 - 2.2340	2.1275	5
7	2.2345 - 2.4480	2.3415	3
8	2.4485 - 2.6620	2.5556	1
9	2.6625 - 2.8760	2.7697	1
10	2.8765 - 3.0900	2.9838	0
11	3.0905 - 3.3050	3.1979	2
12	3.3055 - 3.5190	3.4140	0
13	3.5195 - 3.7330	3.6261	1
14	3.7335 - 3.9470	3.8402	1
15	3.9475 - 4.1610	4.0543	0
16	4.1615 - 4.3750	4.2684	0
17	4.3755 - 4.5890	4.4825	1
18	4.5895 - 4.8030	4.6965	2
19	4.8035 - 5.0170	4.9106	0
20	5.0175 - 5.2310	5.1247	0
21	5.2310 - 5.4450	5.3388	0
22	5.4455 - 5.6600	5.5529	0

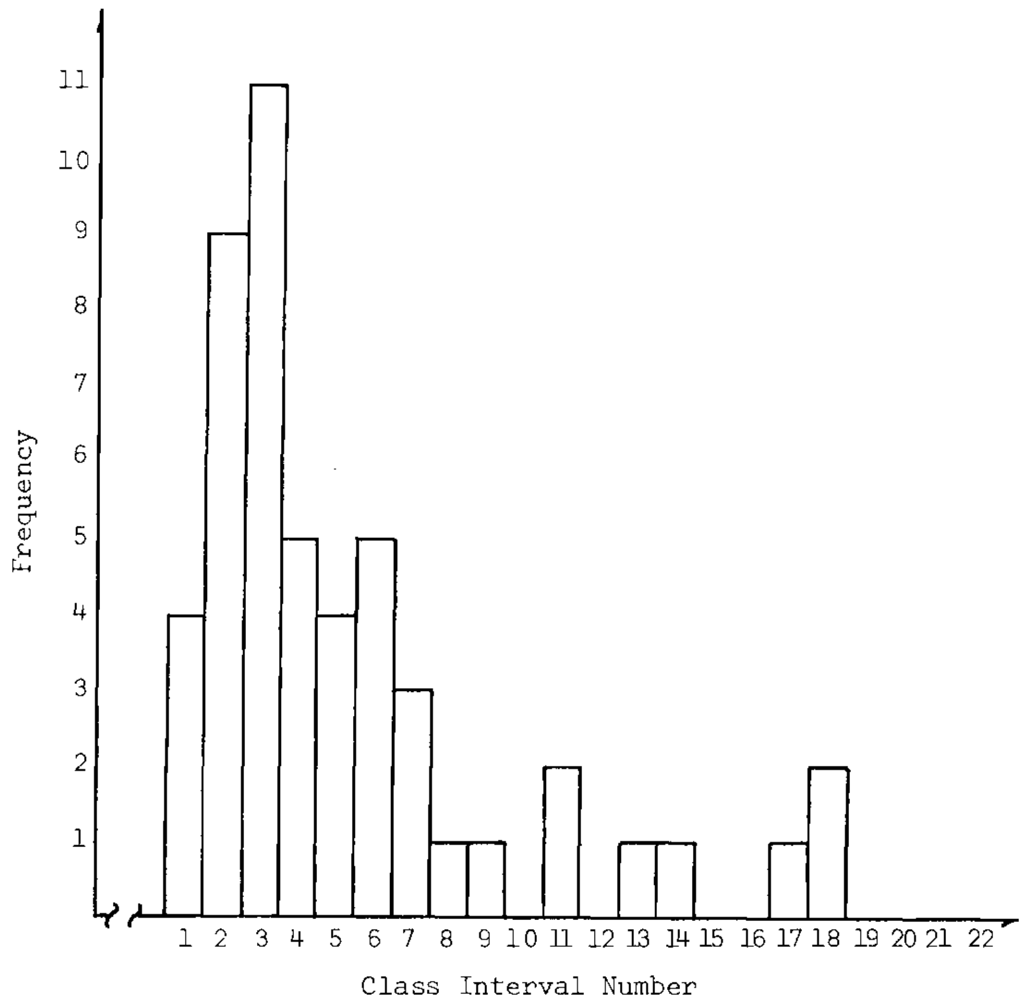

Figure 1. Histogram of Results of Feasible Solution Problem for 50 Sets of Legislative Districts Selected at Random

Fitting a Distribution

The shape of the histogram indicated that a Weibull distribution could be fitted to the data. To check the validity of this assumption the data were plotted on Weibull paper. The results, exhibited in Figure 2, indicated that the assumption was valid.

The parameters of the Weibull distribution fitting the data were estimated so as to minimize the chi-square test statistic, given by:

$$\chi_0^2 = \sum_{i=1}^{k} \frac{(O_i - nE_i)^2}{nE_i}$$

where:

O_i - actual frequency in ith class interval.

nE_i - expected frequency in ith class interval.

k - number of class intervals.

Although the above-mentioned method of parameter estimation has, to my knowledge, not had wide usage, Fisher (3) has shown that, for large samples, the properties of such estimates are the same as the properties of maximum likelihood estimates.

In addition to the desirable properties of maximum likelihood estimates obtained by the use of the described estimator, the estimator itself will yield parameters that tend to make the observed and expected frequencies for each class interval the same. This can be seen by observing that the numerator of χ_0^2 is equivalent to the criterion used to estimate parameters by the method of least squares. This tendency

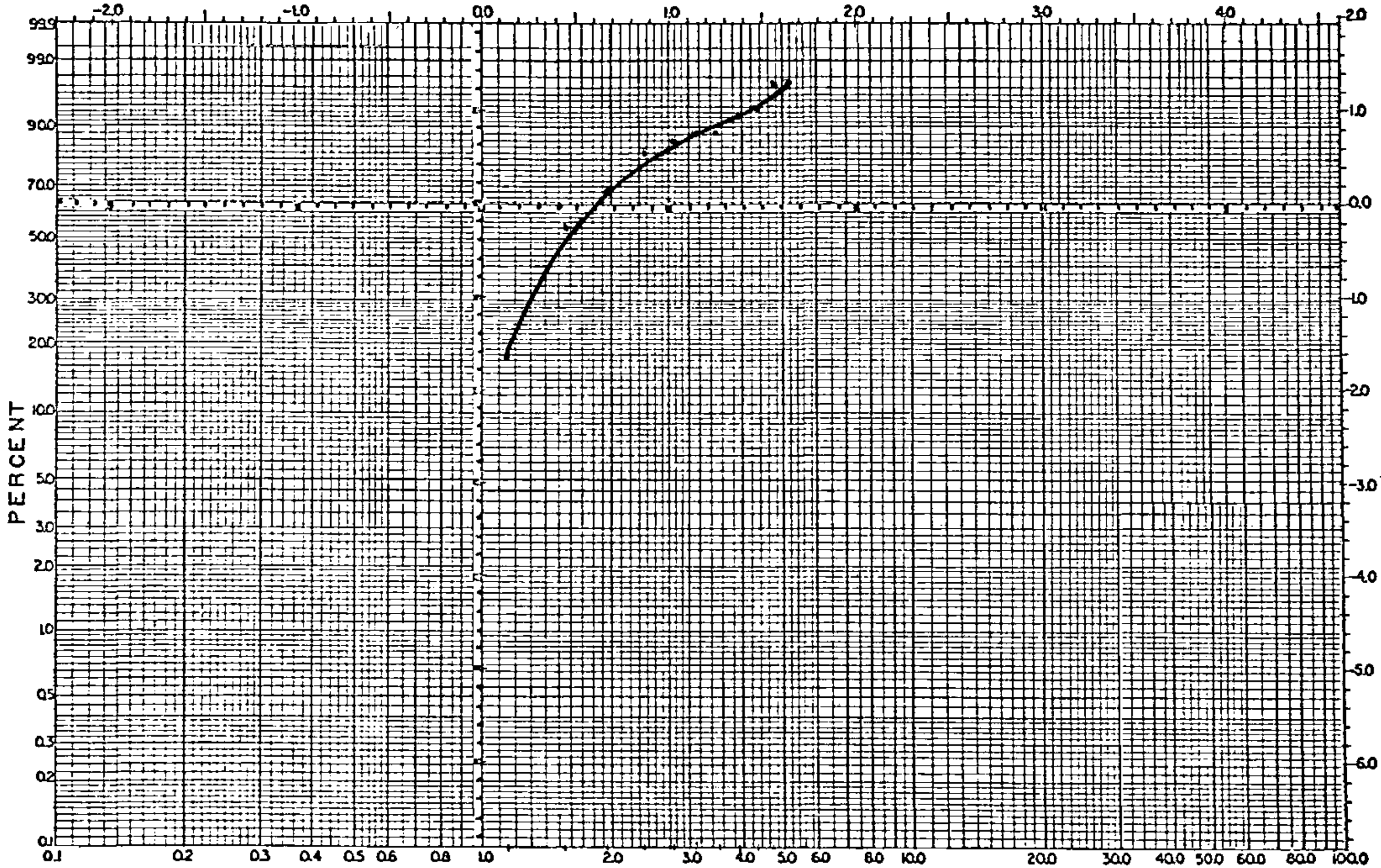

Figure 2. Results of Feasible Solution Problem Plotted on Weibull Paper

Note that by shifting this curve over by approximately one unit the resulting plot is nearly linear. This implies that there is reason to believe that the data are Weibull distributed. For a complete discussion see Reference (2).

of the estimator is very desirable because, theoretically, the chi-square goodness of fit test statistic only has a chi-square distribution when the observed and expected frequencies for each class interval are not significantly different.

The procedure used to minimize χ_0^2 was a directed search method based on the pattern search developed by Hooke and Jeeves (5). Before the minimization was effectuated, the problem was restated as follows:

$$\text{Minimize } \chi_0^2 = \sum_{i=1}^{k} \frac{\{0_i - n[F(x_i)-F(x_{i-1})]\}^2}{n[F(x_i)-F(x_{i-1})]}$$

$$\text{Subject to: } \quad x_i - \gamma \geq 0 \qquad\qquad i = 1,k$$

where:

$$F(x_i) = \int_{\gamma}^{x_i} \frac{\beta(x-\gamma)^{\beta-1}}{\alpha} \exp(-(x-\gamma)^{\beta}/\alpha)dx = 1 - \exp(-(x_i-\gamma)^{\beta}/\alpha)$$

x_i - upper limit of the ith class interval.

0_i - observed frequency in ith class interval.

k - number of class intervals.

N - $\sum_{i=1}^{k} 0_i$.

α - scale parameter of Weibull Distribution.

β - shape parameter of Weibull Distribution.

γ - location parameter of Weibull Distribution.

The results of this minimization are given in Table 2.

Table 2. Results of Minimizing χ_0^2

Parameters of Weibull Distribution that Minimize χ_0^2 Are:

$\alpha = 1.117734$ $\beta = 0.836328$ $\gamma = 1.008789$

Objective Function Value is 16.0319101

Class Interval	Observed Frequency	Expected Frequency
1	4.0000000	3.4900647
2	9.0000000	9.4307220
3	11.0000000	6.5597308
4	5.0000000	5.0639477
5	4.0000000	4.0536679
6	5.0000000	3.3110452
7	3.0000000	2.7406773
8	1.0000000	2.2905230
9	1.0000000	1.9285419
10	0.0000000	1.6334396
11	2.0000000	1.3903018
12	0.0000000	1.1882803
13	1.0000000	1.0192556
14	1.0000000	0.8770149
15	0.0000000	0.7567191
16	0.0000000	0.6545450
17	1.0000000	0.5674352
18	2.0000000	0.4929205
19	0.0000000	0.4289892
20	0.0000000	0.3739902
21	0.0000000	0.3265600
22	0.0000000	0.2855654

Goodness of Fit Test

To test whether it was possible to claim that the data came from a population that could be characterized by a Weibull Distribution, the chi-square goodness of fit test was used. The results are exhibited in Table 3.

Table 3. Chi-square Goodness of Fit Test

Null hypothesis:	Empirical data comes from a population distribution that can be characterized by a Weibull distribution with parameters as follows: $\alpha = 1.117734$ $\beta = 0.836328$ $\gamma = 1.008789$
Alternate hypothesis:	Empirical data does not come from a population that can be characterized by the above specified Weibull distribution.
Level of significance:	$\alpha = 0.05$
Nature of experiment:	Random sampling.
Test statistic:	$\chi_0^2 = \sum_{i=1}^{k} \frac{(O_i - nE_i)^2}{nE_i} = 16.0319101.$
Degrees of freedom:	$V = k - c - 1 = 18.$ c is the number of parameters that were estimated from the data.
Critical value:	$\chi_{\alpha,v}^2 = 28.9.$
Conclusion:	Since $\chi_0^2 < \chi_{\alpha,v}^2$ there is no reason to reject the null hypothesis.

The conclusion from the results of the test indicated that there was no reason to reject the null hypothesis. Although, theoretically, this does not allow us to claim that the data is actually Weibull distributed, the consideration that the null hypothesis would only be rejected at a level of significance below 50 per cent supports, at least intuitively, the assumption that the data was actually Weibull distributed.

<u>Lower Bound Estimation</u>

Figure 3 shows a plot of the expected frequencies of the fitted distribution superimposed on the original histogram. The location parameter of the fitted distribution indicates where the expected frequencies become greater than zero. The units of the location parameter will be the same as the units of the values of the optimal solutions to the feasible solution algorithm.

A lower bound on the values of the objective function of the feasible solution problem may be defined as the lowest value that the location parameter γ of a Weibull distribution may assume before the hypothesis that the data is Weibull distributed must be rejected. If we denote this value of γ as γ^*, then the above may be restated as:

$$\gamma^* = \text{minimum} \{ \underset{\alpha,\beta}{\text{minimum}}\ f(\alpha,\beta,\gamma) \}$$

$$\text{Subject to:} \quad \underset{\alpha,\beta}{\text{minimum}}\ f(\alpha,\beta,\gamma^*) \leq \chi^2_{\alpha,v}$$

where:

$f(\alpha,\beta,\gamma)$ - chi-square test statistic as a function of the parameters α, β, γ.

$f(\alpha,\beta,\gamma^*)$ - Chi-square test statistic as a function of the parameters α and β, γ^* is constant.

$\chi^2_{\alpha,v}$ - critical value of chi-square goodness of fit test.

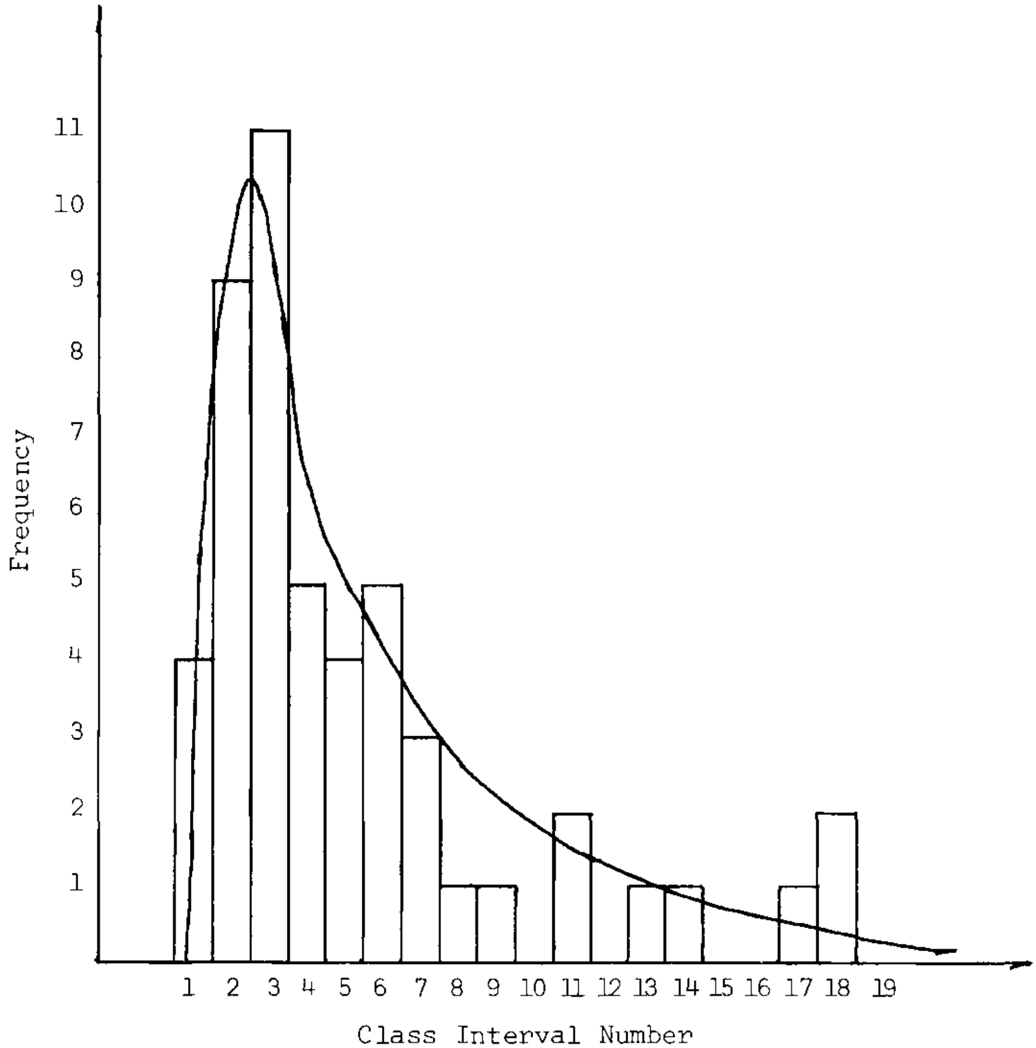

Figure 3. Plot of Actual versus the Expected Frequencies Obtained by Estimating the Parameters of the Weibull Distribution by Minimizing χ_0^2

The above formulation simply states that γ^* is the location parameter of the most extreme Weibull distribution which could fit the data as determined by the chi-square goodness of fit test. By most extreme it is meant that γ^* is the lowest point on the class interval axis from which a Weibull distribution could characterize the population distribution from which the empirical data was obtained.

Since the above interpretation of γ^* implies that the expected frequency of obtaining values of the objective function of the feasible solution algorithm below γ^* is zero, γ^* represents a lower bound on the objective function values of the feasible solution algorithm.

The procedure used to find γ^* is shown in Flowchart form in Figure 4. The initial values of α, β, and γ should be those estimated by minimization of the chi-square test statistic. After each reduction of γ, the parameters α and β should be re-estimated by finding $f^*(\alpha,\beta)$.

This quantity is the objective function value obtained by solving the following problem:

$$\text{Minimize } f(\alpha,\beta) = \sum_{i=1}^{k} \frac{\{O_i - n[F(X_i)-F(X_{i-1})]\}^2}{n[F(X_i)-F(X_{i-1})]}$$

$$\text{Subject to:} \quad X_i - \gamma \geq 0 \qquad\qquad i = 1,k$$

where:

$f(\alpha,\beta)$ - chi-square test statistic as a function of the parameters α and β.

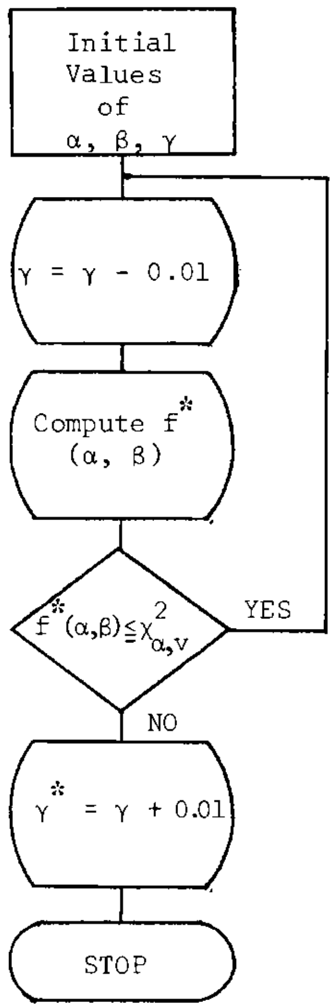

Figure 4. Flowchart of Procedure Followed to Find γ^*

γ - constant.

Other terms - as previously defined.

The conditional block in the flow diagram represents the chi-square goodness of fit test made to determine whether the hypothesis that the population distribution of the data could be considered Weibull with the given parameters. The degrees of freedom for this test, v, are now k-c because the parameter γ is constant and is not estimated from the empirical data.

As soon as the chi-square goodness of fit test results in the rejection of the null hypothesis, the procedure is terminated and γ^* is the last value of γ for which the null hypothesis was not rejected.

The above procedure applied to the test problem yielded a value of γ^* equal to 0.51. This result was obtained by using a level of significance of 5 per cent for the chi-square goodness of fit tests.

CHAPTER IV

CONCLUSIONS AND RECOMMENDATIONS

In the previous chapter a procedure was presented which allowed the estimation of a lower bound to the objective function value of the redistricting problem. The procedure itself consisted of gathering information about the values of the objective function for various initial Legislative District centers and, from this, inferring what the lowest possible objective function value could be. For the particular problem examined, the result appears to be a good estimate of the lower bound, although at this time it is impossible to speculate whether this is a conservative or liberal estimate of the actual solution value.

In addition to the estimation procedure, the minimum chi-square method of parameter estimation appears to hold promise as a fast and efficient estimator. In particular, this method seems to be useful in cases where the maximum likelihood estimates of a distribution are difficult to obtain, but where the properties of such estimates are desirable.

Discussion of the Choice of the Level of Significance

The actual value of γ^* obtained by the procedure described in the previous chapter dependent on the choice of the level of significance used in the chi-square goodness of fit tests. This is so because γ^* will always be the value of the location parameter which makes the value of the chi-square test statistic equal to the critical value of the test.

Since the critical value is dependent on the level of significance used, so is γ^*.

Table 4 exhibits the values of γ^* obtained for various choices of the level of significance.

Table 4. Values of γ^* as a Function of the Level of Significance Used in the Chi-square Goodness of Fit Tests

Level of Significance	γ^*
.25	.80
.10	.65
.05	.51
.01	.27

The interpretation of the relationship between the level of significance used in the chi-square goodness of fit test and the corresponding value of γ^* is not clear. My interpretation is based on the assumption made earlier that the null hypothesis is true if the results of the goodness of fit test failed to reject it. If the null hypothesis is true, then the level of significance indicates the probability of making a Type I error. A Type I error will be made in this case if χ_0^2 is greater than $\chi^2_{\alpha,v}$ because then we have rejected a true hypothesis. Since the relationship between γ and χ_0^2 is such that, as γ decreases χ_0^2 increases, γ^* will be the value of γ that makes the quantity χ_0^2 equal to $\chi^2_{\alpha,v}$. But, if the null hypothesis is true, then χ^2_0 has a chi-square distribution with v degrees of freedom and the probability that there is a value of γ which makes χ^2_0 greater than $\chi^2_{\alpha,v}$ is α. Therefore, one possible manner in which the level of significance of the chi-squared goodness of fit tests and the corresponding obtained value of γ^* may be related is that α is related to the probability that there is a lower value of the estimate of the lower bound than γ^*.

I would recommend that further research be directed to establishing the exact relationship between the various levels of significance and the corresponding values of γ^*.

Discussion of Data Classification

All the results presented thus far were based on the data being classified into 22 class intervals. Since this was a rather arbitrary selection, the procedure was applied to various classifications of data. Table 5 shows the results obtained.

An analysis of variance was made on these results and the conclusions were that there was a significant difference between the lower bounds obtained for various classifications of the data.

Since there seem to be no good quantitative criteria by which to determine into how many class intervals data should be classified, and since, in this case, the results would have been statistically different, I would recommend that further research be directed to establishing criteria for data classification.

Table 5. Levels of Significance and Lower Bounds for Various Number of Class Intervals

Number of Class Intervals	.25	.10	.05	.01
10	.36	–	–	–
13	.94	.84	.77	.61
16	.88	.78	.71	.53
19	.95	.85	.78	.62
22	.80	.65	.54	.27
25	.88	.76	.67	.48
28	.91	.76	.66	.41
31	.96	.82	.72	.49

APPENDIX I

APPENDIX I

ALGORITHM FOR ESTIMATING THE OPTIMAL OBJECTIVE FUNCTION VALUE OF AN OPTIMIZATION PROBLEM

Consider the problem:

$$\text{Minimize } f(X)$$

$$\text{Subject to: } \quad g_j(X) \geq 0 \qquad j = 1,m$$

where:

X - L dimensional vector of decision variables.

$f(X)$ - objective function

$g_j(X)$ - m dimensional constraint set

If the problem is of such nature that it is not practical to find the optimal solution, but if it is practical to obtain and evaluate $f(X)$ for any feasible vector of decision variables X, then an estimate of the optimal value of the objective function may be obtained by the following procedure:

1. Select Z sets of L random numbers. Let each set represent a vector of decision variables, denoted by X.

2. Check each vector obtained in Step One for feasibility. If it is feasible, compute $f(X)$, otherwise discard the vector.

3. Classify the value of f(X) into K intervals and construct a Histogram.

4. Select a probability density function p[f(X)] to characterize the population distribution of the values of f(X). This density function should have a lower bound in the form of a location parameter.

5. Denote the location parameter of p[f(X)] by γ and the remaining parameters by θ_i.

6. Estimate the values of γ and θ_i by minimizing the quantity:

$$X_0^2 = \sum_{i=1}^{k} \frac{(O_i - nE_i)^2}{NE_i}$$

where:

O_i - observed frequency of values of f(X) in the ith class interval.

E_i - probability that $C_i \leq f(X) \leq C_{i+1}$.

C_i - upper limit of ith class interval.

n - $\sum_{i=1}^{k} O_i$

k - number of class intervals.

7. Perform the chi-square goodness of fit test as follows:

Null hypothesis: Population distribution of f(X) may be characterized by p[f(X)] with the parameters estimated in Step 6.

Alternate hypothesis: Population distribution of f(X) may not be characterized by p[f(X)] with the parameters estimated in Step 6:

Degrees of freedom:

$$V = K - C - 1$$

where:

V - degrees of freedom.

K - number of class intervals.

C - number of parameters of p[f(X)].

Conclusions: If $\chi_0^2 \leq \chi_{\alpha,v}^2$ then fail to reject the null hypothesis. Otherwise reject the null hypothesis.

8. If the conclusion of Step 7 fails to reject the null hypothesis, then an estimate of the objective function value to the optimization problem is given by γ^*, where:

$$\gamma^* = \underset{\gamma}{\text{minimize}} \{\underset{\theta_i}{\text{minimum}}\ \chi_0^2\ (\theta_i)\}$$

where:

$\chi_0^2\ (\theta_i)$ - the chi-square test statistics as a function of the parameters θ_i.

γ - location parameter of p[f(X)].

9. If the conclusion to Step 7 was to reject the null hypothesis, then a different function p[f(X)] must be considered to represent the population distribution of f(X). With the new p[f(X)] the procedure should be started at Step 5 again.

This algorithm is exhibited in Flowchart form in Figure 5.

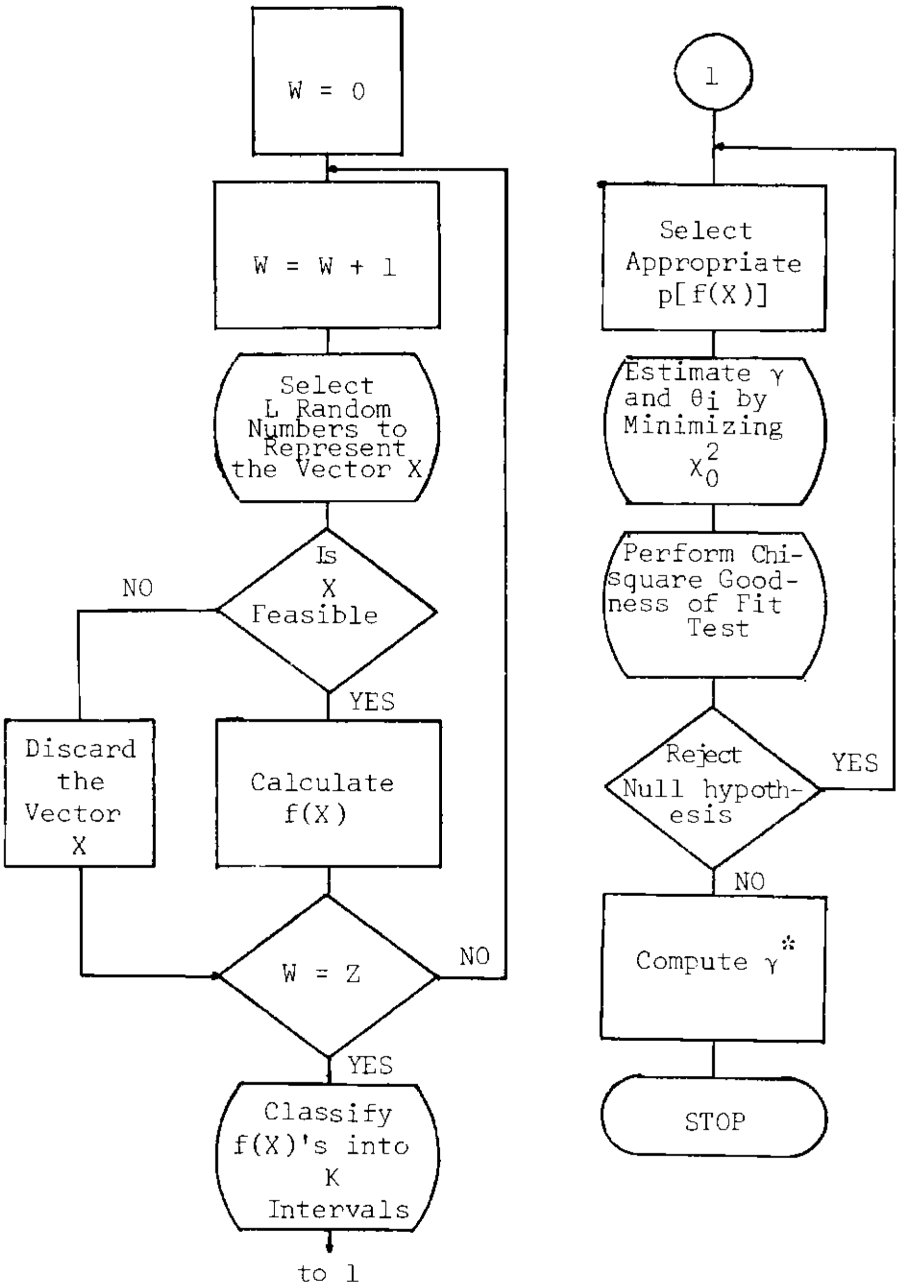

Figure 5. Flowchart of Algorithm to Estimate the Optimal Objective Function Value of an Optimization Problem

APPENDIX II

APPENDIX II

COMPUTER PROGRAM

The following is the listing of the computer program used to:

1. Classify data obtained from the solutions to the allocation problem.

2. Find the parameters of the Weibull Distribution fitting the data by minimizing the chi-square test statistic for goodness of fit.

```
GEORGIA TECH HECC B-5500 ALGOL COMPILER     TUESDAY,   6/13/67,  1106 AM.
    BEGIN                                                                    0000
                                                         START OF SEGMENT **********    2
    FILE IN CRDS (2,10);                                                     0000
    FILE OUT PRTC 4(2,15);                                                   0003
    FILE OUT WNS 0(2,10);                                                    0007
    INTEGER O,I,NINTER,K,N;                                                  0010
    REAL SUM,AVG,DIFF,SS,VAR,SQRDOT,SSUM,DIFFS,RANGE,SPHI;                   0010
    REAL DEL,RHO,DELTA,SPSI,THETA;                                           0010
    REAL KEEP1,KEEP2,KEEP3,KEEP4;                                            0010
    REAL ARRAY T,Z,MID,FREQ[0:100],PHI,PSI[0:10];                            0010
    REAL ARRAY X,Y[0:100];                                                   0015
         LIST RAWDATA(T[I]);                                                 0017
         FORMAT FORM1(I2);                                                   0023
                                                         START OF SEGMENT **********    3
                                                           3 IS    4 LONG, NEXT SEG     2
         FORMAT FORM2(R8.0);                                                 0023
                                                         START OF SEGMENT **********    4
                                                           4 IS    4 LONG, NEXT SEG     2
         FORMAT FORM3(X10,"THE FIRST MOMENT IS",X3,F10.2,/);                 0023
                                                         START OF SEGMENT **********    5
                                                           5 IS   11 LONG, NEXT SEG     2
         FORMAT FORM4(X10,"THE SECOND MOMENT IS",X3,E20.11,/);               0023
                                                         START OF SEGMENT **********    6
                                                           6 IS   11 LONG, NEXT SEG     2
         FORMAT FORM5(X10,"THE STANDARD DEVIATION IS",X3,F10.2,/);           0023
                                                         START OF SEGMENT **********    7
                                                           7 IS   12 LONG, NEXT SEG     2
         FORMAT FORM6(///,X10,"SORTED VALUES");                              0023
                                                         START OF SEGMENT **********    8
                                                           8 IS   10 LONG, NEXT SEG     2
         FORMAT FORM7(///,X10,"INPUT VALUES");                               0023
                                                         START OF SEGMENT **********    9
                                                           9 IS    9 LONG, NEXT SEG     2
         FORMAT FORM8(X10,"TOTAL IS EQUAL TO",X3,F15.2,/);                   0023
                                                         START OF SEGMENT **********   10
                                                          10 IS   10 LONG, NEXT SEG     2
PROCEDURE TIEMPO;                                                            0023
BEGIN                                                                        0023
REAL ARRAY T[0:2];                                                           0023
                                                         START OF SEGMENT **********   11
FORMAT FMT1(///,"PROCESSOR TIME=",F9.2,/,"I O TIME=",F9.2,/,                 0001
                                                         START OF SEGMENT **********   12
            "TOTAL TIME=",F9.2,///);                                         0001
                                                          12 IS   21 LONG, NEXT SEG    11
    T[0]←TIME(2)/60.;                                                        0001
    T[1]←TIME(3)/60.;                                                        0004
    T[2]←T[0]+T[1];                                                          0006
    WRITE(PRTC,FMT1,T[0],T[1],T[2]);                                         0008
END OF TIEMPO;                                                               0021
                                                          11 IS   25 LONG, NEXT SEG     2
    PROCEDURE SORT;                                                          0023
    BEGIN                                                                    0023
         INTEGER IK,K;                                                       0023
                                                         START OF SEGMENT **********   13
         REAL MAX;                                                           0000
         FORMAT FORM9(//,X10,"THE RANGE OF THIS DATA IS",X3,F10.2,///);      0000
                                                         START OF SEGMENT **********   14
                                                          14 IS   16 LONG, NEXT SEG    13
         FOR IK←1 STEP 1 UNTIL O DO                                          0000
    BEGIN                                                                    0001
```

```
        MAX←T[1];                                                           0001
        K←1;                                                                0002
        FOR I←1 STEP 1 UNTIL N DO                                           0002
    BEGIN                                                                   0004
        IF T[I]>MAX THEN BEGIN                                              0004
        MAX←T[I];                                                           0005
        K←I;                                                                0006
    END;                                                                    0007
    END;                                                                    0007
        Z[IK]←T[K];                                                         0009
        T[K]←0;                                                             0011
    END;                                                                    0012
        WRITE(PRTC,FORM6);                                                  0014
        FOR I←1 STEP 1 UNTIL N DO                                           0017
    BEGIN                                                                   0019
        WRITE(PRTC,FORM2,Z[I]);                                             0019
    END;                                                                    0027
        RANGE←Z[1]-Z[0];                                                    0029
        WRITE(PRTC,FORM9,RANGE);                                            0031
TIEMPO;                                                                     0039
    END OF SORT;                                                            0039
                                                     13 IS   42 LONG, NEXT SEG     2
        PROCEDURE HISTOG;                                                   0023
    BEGIN                                                                   0023
        REAL INLONG,SSSUM;                                                  0023
                                                   START OF SEGMENT **********    15
        REAL ARRAY POINT[0:100],VALUR[0:100,0:100],RLFREQ[0:100],          0000
                                  CUFREQ[0:100];                            0005
        INTEGER M,J,LL;                                                     0007
        INTEGER ARRAY MM[0:100];                                            0007
        LABEL COST;                                                         0009
        FORMAT FORM10(X45,"HISTOGRAM DATA",//);                             0009
                                                   START OF SEGMENT **********    16
                                                     16 IS    9 LONG, NEXT SEG    15
        FORMAT FORM11(X10,F10.2,"-",F10.2,X20,I5,X11,F7.4,X7,F8.4);         0009
                                                   START OF SEGMENT **********    17
                                                     17 IS   13 LONG, NEXT SEG    15
        FORMAT FORM12(X35,F11.2,//);                                        0009
                                                   START OF SEGMENT **********    18
                                                     18 IS    7 LONG, NEXT SEG    15
        FORMAT FORM13(X12,"CLASS BOUNDARIES",X7,"CLASS MEMBERS",            0009
                                                   START OF SEGMENT **********    19
        X3,"FREQUENCY",X7,"REL. FREQ.",X5,"CUM.REL.FREQ.",//);             0009
                                                     19 IS   23 LONG, NEXT SEG    15
        FORMAT FMT14(X9,F10.2,X3,F10.2,/);                                  0009
                                                   START OF SEGMENT **********    20
                                                     20 IS    8 LONG, NEXT SEG    15
        FORMAT FMT15(///,X16,"SUMMARY",//,X10,"MIDPOINTS",                  0009
                                                   START OF SEGMENT **********    21
                    X3,"FREQUENCY",/);                                      0009
                                                     21 IS   18 LONG, NEXT SEG    15
    FORMAT FMT16(F12.6,",",F12.6,",");                                      0009
                                                   START OF SEGMENT **********    22
                                                     22 IS    7 LONG, NEXT SEG    15
    FORMAT FMT17("000000000000000000000000000000000000000000000000");       0009
                                                   START OF SEGMENT **********    23
                                                     23 IS   12 LONG, NEXT SEG    15
    FORMAT FMT18(I3,",");                                                   0009
                                                   START OF SEGMENT **********    24
                                                     24 IS    5 LONG, NEXT SEG    15
        INLONG←RANGE/NINTER;                                                0009
        POINT[0]←Z[0];                                                      0010
```

```
        FOR I←1 STEP 1 UNTIL NINTER DO                                  0011
    BEGIN                                                               0013
        POINT[I]←POINT[I-1]+INLONG     ;                                0013
        MID[I]←POINT[I]-INLONG/2;                                       0015
        MID[I]←MID[I]/1000000;                                          0018
    END;                                                                0020
        SSSUM←0;                                                        0022
        FOR J←1 STEP 1 UNTIL NINTER DO                                  0023
    BEGIN                                                               0025
        IF J=NINTER THEN BEGIN                                          0025
        M←0;                                                            0026
        FOR I←N STEP-1 UNTIL 1 DO                                       0027
    BEGIN                                                               0028
        IF Z[I]≤POINT[J] AND Z[I]≥POINT[J-1] THEN                       0028
    BEGIN                                                               0031
        M←M+1;                                                          0031
        VALUR[J,M]←Z[I];                                                0033
    END;                                                                0035
    END;                                                                0035
        GO TO COST;                                                     0037
    END;                                                                0038
        M←0;                                                            0038
        FOR I←N STEP -1 UNTIL 1 DO                                      0038
    BEGIN                                                               0040
        IF Z[I]<POINT[J] AND Z[I]≥POINT[J-1] THEN                       0040
    BEGIN                                                               0043
        M←M+1;                                                          0043
        VALUR[J,M]←Z[I];                                                0045
    END;                                                                0047
    END;                                                                0047
        COST:MM[J]←M;                                                   0049
        FREQ[J]←MM[J];                                                  0051
        RLFREQ[J]←MM[J]/O;                                              0052
        SSSUM←SSSUM+RLFREQ[J];                                          0054
        CUFREQ[J]←SSSUM;                                                0056
    END;                                                                0057
        WRITE(PRTC,FORM10);                                             0059
        WRITE(PRTC,FORM13);                                             0062
        FOR J←1 STEP 1 UNTIL NINTER DO                                  0065
    BEGIN                                                               0067
        WRITE(PRTC,FORM11,POINT[J-1],POINT[J],FREQ[J],RLFREQ[J],        0067
              CUFREQ[J]);                                               0077
        LL←MM[J];                                                       0081
        FOR I←1 STEP 1 UNTIL LL DO                                      0082
    BEGIN                                                               0083
        WRITE(PRTC,FORM12,VALUR[J,I]);                                  0083
    END;                                                                0091
    END;                                                                0093
        WRITE(PRTC[PAGE]);                                              0095
        WRITE(PRTC,FMT15);                                              0098
        WRITE(WWS,FMT17);                                               0101
        WRITE(WWS,FMT18,NINTER);                                        0104
        FOR I←1 STEP 1 UNTIL NINTER DO                                  0112
    BEGIN                                                               0113
        WRITE(PRTC,FMT14,MID[I],RLFREQ[I]);                             0113
        WRITE(WWS,FMT16,MID[I],RLFREQ[I]);                              0122
    END;                                                                0132
TIEMPO;                                                                 0134
        WRITE(PRTC[PAGE]);                                              0135
    END OF HISTOG;                                                      0137
                                               15 IS  144 LONG, NEXT SEG     2
  PROCEDURE DETAIL;                                                     0023
```

```
BEGIN                                                                     0023
    REAL A,B,G,SUM,ADD,QUAN;                                              0023
                                                     START OF SEGMENT **********   25
    REAL ARRAY FX[0:100];                                                 0000
FORMAT FMT1(X30,"WEIBULL EVALUATOR IN CONJUNCTION WITH SEARCH"            0001
                                                     START OF SEGMENT **********   26
            " TECHNIQUES",/,X40,"PARAMETERS ARE: A=",F12.6,X2,"B=",       0001
             F12.6, X2,"G=",F12.6,//,X10,"INTERVAL",X4,"OBSERVED",        0001
             X4,"EXPECTED",X4,"CHI-SQR.",X4,"CUMM CHI-SQR." ,/);          0001
                                                       26 IS   45 LONG, NEXT SEG   25
    FORMAT FMT2(X8,I12,4F12.7);                                           0001
                                                     START OF SEGMENT **********   27
                                                       27 IS    6 LONG, NEXT SEG   25
    LABEL LBL;                                                            0001
         WRITE(PRTC[PAGE]);                                               0001
         A←PHI[1];                                                        0004
         B←PHI[2];                                                        0005
         G←PHI[3];                                                        0006
         WRITE(PRTC,FMT1,A,B,G);                                          0007
         SUM←0;                                                           0018
         FOR I←1 STEP 1 UNTIL N DO                                        0019
    BEGIN                                                                 0020
         FX[I]←1.0-EXP(-(X[I]-G)*B/A);                                    0020
         ADD←FX[I]-FX[I-1];                                               0025
         ADD←D×ADD;                                                       0028
         IF ADD=0 THEN BEGIN                                              0029
         QUAN←0;                                                          0030
GO TO LBL;                                                                0031
    END;                                                                  0033
    LBL:                                                                  0033
         QUAN←(Y[I]-ADD)*2/ADD;                                           0033
         SUM←SUM+QUAN;                                                    0035
         WRITE(PRTC,FMT2,I,Y[I],ADD,QUAN,SUM);                            0036
    END;                                                                  0050
TIEMPO;                                                                   0052
    END OF DETAIL;                                                        0053
                                                       25 IS   58 LONG, NEXT SEG    2
  REAL PROCEDURE EVALUATE(PHI,X,Y,N,K);                                   0023
    VALUE PHI,X,Y,N,K;                                                    0023
  REAL ARRAY PHI,X,Y[0];                                                  0023
  INTEGER K,N;                                                            0023
  BEGIN                                                                   0023
    REAL ARRAY FX[0:100] ;                                                0023
                                                     START OF SEGMENT **********   28
    REAL SUM,ADD,QUAN;                                                    0001
    INTEGER I;                                                            0001
    LABEL  THEE,THOU,LBL;                                                 0001
         FX[0]←0;                                                         0001
         D←51;                                                            0003
         IF PHI[2]≤0 THEN GO TO THOU;                                     0003
         SUM←0;                                                           0005
         IF PHI[3]>2 THEN GO TO THOU;                                     0006
         FOR I←1 STEP 1 UNTIL N DO                                        0007
    BEGIN                                                                 0009
         QUAN←X[I]-PHI[3];                                                0009
         IF QUAN<0 THEN                                                   0010
    BEGIN                                                                 0011
    THOU:                                                                 0012
         EVALUATE←100000;                                                 0012
         GO TO THEE;                                                      0012
    END                                                                   0015
    ELSE                                                                  0015
```

```
        IF PHI[1]=0 THEN GO TO THOU;                                            0015
        FX[I]←1.0-EXP(-(X[I]-PHI[3])*PHI[2]/PHI[1]);                            0017
        ADD←FX[I]-FX[I-1];                                                      0023
        ADD←D×ADD;                                                              0025
        IF ADD=0 THEN GO TO LBL;                                                0027
        SUM←SUM+(Y[I]-ADD)*2/ADD;                                               0028
    LBL:                                                                        0031
   END;                                                                         0032
        EVALUATE←SUM;                                                           0034
  THEE:                                                                         0035
  END OF EVALUATE;                                                              0035
                                                          28 IS   41 LONG, NEXT SEG     2
  PROCEDURE E;                                                                  0023
  BEGIN                                                                         0023
        FOR I←1 STEP 1 UNTIL K DO                                               0023
   BEGIN                                                                        0025
        PHI[I]←PHI[I]+DEL;                                                      0025
        SPHI←EVALUATE(PHI,X,Y,N,K);                                             0027
        IF SPHI<SS THEN SS←SPHI ELSE                                            0030
   BEGIN                                                                        0032
        PHI[I]←PHI[I]-2×DEL;                                                    0033
        SPHI←EVALUATE(PHI,X,Y,N,K);                                             0035
        IF SPHI<SS THEN SS←SPHI ELSE PHI[I]←PHI[I]+DEL;                         0039
   END;                                                                         0044
   END;                                                                         0044
  END OF E;                                                                     0046
  PROCEDURE INPUT;                                                              0046
   BEGIN                                                                        0046
   FORMAT FMT1(X5,I3,//);                                                       0046
                                                  START OF SEGMENT **********    29
                                                  START OF SEGMENT **********    30
                                                    30 IS    7 LONG, NEXT SEG    29
   FORMAT FMT2(X5,2F15.6);                                                      0000
                                                  START OF SEGMENT **********    31
                                                    31 IS    5 LONG, NEXT SEG    29
   FORMAT FMT3(//,X5,I4,3F15.7,//);                                             0000
                                                  START OF SEGMENT **********    32
                                                    32 IS   10 LONG, NEXT SEG    29
   FORMAT FMT5(X5,F15.7);                                                       0000
                                                  START OF SEGMENT **********    33
                                                    33 IS    5 LONG, NEXT SEG    29
        N←NINTER;                                                               0000
        FOR I←1 STEP 1 UNTIL N DO                                               0000
   BEGIN                                                                        0002
        X[I]←MIO[I];                                                            0002
        Y[I]←FREQ[I];                                                           0003
   END;                                                                         0005
        WRITE(PRTC,FMT1,N);                                                     0007
        FOR I←1 STEP 1 UNTIL N DO                                               0015
   BEGIN                                                                        0016
        WRITE(PRTC,FMT2,X[I],Y[I]);                                             0016
   END;                                                                         0025
        WRITE(PRTC,FMT3,K,DEL,RHO,DELTA);                                       0027
        FOR I←1 STEP 1 UNTIL K DO                                               0039
   BEGIN                                                                        0040
        WRITE(PRTC,FMT5,PSI[I]);                                                0040
   END;                                                                         0048
TIEMPO:                                                                         0050
  END OF INPUT;                                                                 0051
                                                    29 IS   52 LONG, NEXT SEG     2
PROCEDURE OPTIMIZE;                                                             0046
BEGIN                                                                           0046
```

```
COMMENT THIS PROGRAM DOES AN OPTIMUM SEARCH BY THE HOOKE AND JEEVES                    0046
PATTERN SEARCH PROCEDURE- WRITTEN BY W W SWART-SPRING 1967;                            0046
INTEGER NO,T;                                                                          0046
                                                        START OF SEGMENT **********        34
LABEL START,ONE,TWO;                                                                   0000
 FORMAT FMT1(3F12.6,/);                                                                0000
                                                        START OF SEGMENT **********        35
                                                          35 IS    5 LONG, NEXT SEG        34
FORMAT FMT4(//);                                                                       0000
                                                        START OF SEGMENT **********        36
                                                          36 IS    5 LONG, NEXT SEG        34
FORMAT FMT6(///,X30,"PATTERN SEARCH FOR IE 787-WRITTEN BY WW SWART")                   0000
                                                        START OF SEGMENT **********        37
     ;                                                                                 0000
                                                          37 IS   15 LONG, NEXT SEG        34
FORMAT FMT7(X10,"BASE POINT NUMBER",I4,X3,"STEPSIZE",F12.7,//);                        0000
                                                        START OF SEGMENT **********        38
                                                          38 IS   14 LONG, NEXT SEG        34
FORMAT FMT8(X15,"A=",F12.7,"B=",F12.7,"G=",F12.7,X5,"F(A,B,G)=",                       0000
                                                        START OF SEGMENT **********        39
           F12.7,/);                                                                   0000
                                                          39 IS   15 LONG, NEXT SEG        34
FORMAT FMT9(/,X10,"EXPLORATORY MOVE RESULTS",/,X15,"A=",F12.7,                         0000
                                                        START OF SEGMENT **********        40
           "B=",F12.7,"G=",F12.7,X5,"F(A,B,G)=",F12.7);                                0000
                                                          40 IS   21 LONG, NEXT SEG        34
FORMAT FMT10(//,"AFTER",I4,"BASE POINT EXPLORATIONS WE HAVE AT",                       0000
                                                        START OF SEGMENT **********        41
           "LEAST A LOCAL OPTIMUM LOCATED AT:",/,X5,"A=",F12.7,                        0000
           "B=",F12.7,""G=",F12.7,"WITH MINIMUM VALUE OF F(A,B,G)",                    0000
           "=",F12.7);                                                                 0000
                                                          41 IS   34 LONG, NEXT SEG        34
     FORMAT FMT12(3(F12.7,",")));                                                      0000
                                                        START OF SEGMENT **********        42
                                                          42 IS    7 LONG, NEXT SEG        34
INPUT;                                                                                 0000
     NO+0;                                                                             0000
     WRITE(PRTC[PAGE]);                                                                0001
     WRITE(PRTC,FMT1,PSI[1],PSI[2],PSI[3]);                                            0004
     WRITE(PRTC,FMT6);                                                                 0015
     WRITE(PRTC,FMT4);                                                                 0018
START:SPSI+EVALUATE(PSI,X,Y,N,K);                                                      0021
 ONE:SS+SPSI;                                                                          0025
COMMENT ESTABLISH BASE POINT;                                                          0026
     FOR I+1 STEP 1 UNTIL K DO                                                         0026
BEGIN                                                                                  0028
     PHI[I]+PSI[I]              ;                                                      0028
END;                                                                                   0029
     WRITE(PRTC,FMT1,PHI[1],PHI[2],PHI[3]);                                            0031
     NO+NO+1;                                                                          0043
     WRITE(PRTC,FMT7,NO,DEL);                                                          0044
     WRITE(PRTC,FMT8,PHI[1],PHI[2],PHI[3],SS);                                         0054
COMMENT MAKE EXPLORATORY MOVES;                                                        0067
E;                                                                                     0067
     WRITE(PRTC,FMT9,PHI[1],PHI[2],PHI[3],SS);                                         0067
COMMENT ARE EXPLORATORY MOVES SUCCESSFUL;                                              0080
     IF SS<SPSI THEN                                                                   0080
BEGIN                                                                                  0081
COMMENT IF SUCCESSFUL MAKE PATTERN MOVE;                                               0081
 TWO:FOR I+1 STEP 1 UNTIL K DO                                                         0081
BEGIN                                                                                  0083
     THETA+PSI[I];                                                                     0083
```

```
        PSI[I]←PHI[I];                                                          0084
        PHI[I]←2×PHI[I]-THETA;                                                  0085
    END;                                                                        0088
    COMMENT END OF PATTERN MOVE,NEW POINT PHI IS THE NEW BASE POINT;            0090
        SPSI←SS;                                                                0090
        SS←SPHI←EVALUATE(PHI,X,Y,N,K);                                          0091
    COMMENT MAKE EXPLORATORY MOVES;                                             0095
        NO←NO+1;                                                                0095
        WRITE(PRTC,FMT9,PHI[1],PHI[2],PHI[3],SS);                               0096
    E;                                                                          0109
    COMMENT IF SUCCESSFUL MAKE NEW PATTERN MOVE,OTHERWISE GO TO 1 AND           0109
        TRY NEW EXPLORATORY MOVES;                                              0109
        IF SS<SPSI THEN GO TO TWO ELSE GO TO ONE;                               0109
    END                                                                         0111
    ELSE                                                                        0111
    COMMENT IF EXPLORATORY MOVES WERE NOT SUCCESSFUL REDUCE STEP SIZE           0111
        UNTIL THESE ARE < THAN DELTA;                                           0111
        IF DEL> DELTA THEN                                                      0111
    BEGIN                                                                       0112
        DEL←RHO×DEL;                                                            0113
        GO TO ONE;                                                              0114
    END;                                                                        0115
        WRITE(WNS,FMT12,PHI[1],PHI[2],PHI[3]);                                  0115
        WRITE(PRTC,FMT10,NO,PHI[1],PHI[2],PHI[3],SS);                           0126
TIEMPO;                                                                         0140
 DETAIL;                                                                        0140
END OF OPTIMIZE;                                                                0141
                                                  34 IS  144 LONG, NEXT SEG     2
        WRITE(PRTC[NO]);                                                        0046
        READ(CRDS,FORM1,D);                                                     0049
        READ(CRDS,FORM1,NINTER);                                                0057
        READ(CRDS,/,K,DEL,RHO,DELTA);                                           0065
        FOR I←1 STEP 1 UNTIL K DO                                               0077
    BEGIN                                                                       0079
        READ(CRDS,/,PSI[I]);                                                    0079
    END;                                                                        0087
        FOR I←1 STEP 1 UNTIL D DO                                               0090
    BEGIN                                                                       0091
        READ(CRDS,FORM2,RAWDATA);                                               0091
    END;                                                                        0094
        CLOSE(CRDS,RELEASE);                                                    0097
        SUM←0;                                                                  0098
        FOR I←1 STEP 1 UNTIL D DO                                               0099
    BEGIN                                                                       0101
        SUM←SUM+T[I];                                                           0101
    END;                                                                        0102
        AVG←SUM/D;                                                              0104
        SSUM←0;                                                                 0106
        FOR I←1 STEP 1 UNTIL D DO                                               0106
    BEGIN                                                                       0108
        DIFF←T[I]-AVG;                                                          0108
        DIFFS←DIFF*2;                                                           0109
        SSUM←SSUM+DIFFS;                                                        0110
    END;                                                                        0112
        VAR←SSUM/(D-1);                                                         0114
        SQRDOT←SQRT(VAR);                                                       0116
        WRITE(PRTC,FORM8,SUM);                                                  0117
        WRITE(PRTC,FORM3,AVG);                                                  0125
        WRITE(PRTC,FORM4,VAR);                                                  0133
        WRITE(PRTC,FORM5,SQRDOT);                                               0141
        WRITE(PRTC,FORM7);                                                      0149
        FOR I←1 STEP 1 UNTIL D DO                                               0152
```

```
            BEGIN                                                                          0153
                WRITE(PRTC,FORM2,RAWDATA);                                                 0153
            END;                                                                           0156
                SORT;                                                                      0158
                KEEP1←PSI[1];                                                              0159
                KEEP2←PSI[2];                                                              0160
                KEEP3←PSI[3];                                                              0161
                KEEP4←DEL;                                                                 0162
                FOR NINTER←NINTER STEP 3 UNTIL 31 DO                                       0162
            BEGIN                                                                          0164
                HISTOG;                                                                    0164
                PSI[1]←KEEP1;                                                              0164
                PSI[2]←KEEP2;                                                              0165
                PSI[3]←KEEP3;                                                              0167
                DEL←KEEP4;                                                                 0168
                OPTIMIZE;                                                                  0169
        TIEMPO;                                                                            0169
                WRITE(PRTC[PAGE]);                                                         0170
            END;                                                                           0172
        END.                                                                               0175
                                                                   2 IS  178 LONG, NEXT SEG    1
EXP IS SEGMENT NUMBER 0043,PRT ADDRESS IS 0136
LN IS SEGMENT NUMBER 0044,PRT ADDRESS IS 0135
SQRT IS SEGMENT NUMBER 0045,PRT ADDRESS IS 0202
OUTPUT(W) IS SEGMENT NUMBER 0046,PRT ADDRESS IS 0105
BLOCK CONTROL IS SEGMENT NUMBER 0047,PRT ADDRESS IS 0005
INPUT(W) IS SEGMENT NUMBER 0048,PRT ADDRESS IS 0176
X TO THE I IS SEGMENT NUMBER 0049,PRT ADDRESS IS 0137
ALGOL WRITE   IS SEGMENT NUMBER 0050,PRT ADDRESS IS 0014
ALGOL READ    IS SEGMENT NUMBER 0051,PRT ADDRESS IS 0015
ALGOL SELECT  IS SEGMENT NUMBER 0052,PRT ADDRESS IS 0016
                                                                   1 IS    2 LONG, NEXT SEG    0
                                                                  53 IS   69 LONG, NEXT SEG    0
NUMBER OF ERRORS DETECTED =   0.  COMPILATION TIME =   59 SECONDS.
PRT SIZE = 135; TOTAL SEGMENT SIZE =  1159 WORDS; DISK SIZE =  73 SEGS; NO. PGM. SEGS =  53
ESTIMATED CORE STORAGE REQUIREMENT =  6875 WORDS.
```

The following is a listing of the program used to do the operations shown in Figure 4.

```
GEORGIA TECH RECC B-5500 ALGOL COMPILER      FRIDAY,   6/16/67, 11:22 AM.

BEGIN                                                                        10000100    0000
                                                                START OF SEGMENT **********    2
    COMMENT THIS PROGRAM DOES AN OPTIMUM SEARCH BY THE HOOKE AND JEEVES      10000200    0000
    PATTERN SEARCH PROCEDURE- WRITTEN BY W W SWART-SPRING 1967;              10000300    0000
    FILE IN CRDS(2,10);                                                      10000400    0000
    FILE OUT WWS 0(2,10);                                                    10000500    0003
    FILE OUT PRNT 6(2,15);                                                   10000600    0007
    INTEGER NU,I,K,N;                                                        10000700    0010
    REAL DEL,SPHI,SS,RHO,DELTA,SPST,THETA,U,KEEP;                            10000800    0010
    REAL ARRAY X,Y[0:100],PHI,PSI[0:10],T[0:2];                              10000900    0010
    REAL ARRAY STOR[0:10];                                                   10001000    0016
    LABEL START,ONE,TWO,FINAL,ITER;                                          10001100    0018
    FORMAT FMT1(X5,I3,//);                                                   10001200    0018
                                                                START OF SEGMENT **********    3
                                                                  3 IS    7 LONG, NEXT SEG     2
    FORMAT FMT2(X5,2F15.6);                                                  10001300    0018
                                                                START OF SEGMENT **********    4
                                                                  4 IS    5 LONG, NEXT SEG     2
    FORMAT FMT3(//,X5,I4,3F15.7,//);                                         10001400    0018
                                                                START OF SEGMENT **********    5
                                                                  5 IS   10 LONG, NEXT SEG     2
    FORMAT FMT4(//);                                                         10001500    0018
                                                                START OF SEGMENT **********    6
                                                                  6 IS    5 LONG, NEXT SEG     2
    FORMAT FMT5(X5,F15.7);                                                   10001600    0018
                                                                START OF SEGMENT **********    7
                                                                  7 IS    5 LONG, NEXT SEG     2
    FORMAT FMT6(///,X30,"PATTERN SEARCH FOR IE 787-WRITTEN BY WW SWART")     10001700    0018
                                                                START OF SEGMENT **********    8
         ;                                                                   10001800    0018
                                                                  8 IS   15 LONG, NEXT SEG     2
    FORMAT FMT7(X10,"BASE POINT NUMBER",I4,X3,"STEPSIZE",F12.7,//);          10001900    0018
                                                                START OF SEGMENT **********    9
                                                                  9 IS   14 LONG, NEXT SEG     2
    FORMAT FMT8(X15,"A=",F12.7,"B=",F12.7,"G=",F12.7,X5,"F(A,B,G)=",         10002000    0018
                                                                START OF SEGMENT **********   10
              F12.7,/);                                                      10002100    0018
                                                                 10 IS   15 LONG, NEXT SEG     2
    FORMAT FMT9(/,X10,"EXPLORATORY MOVE RESULTS",/,X15,"A=",F12.7,           10002200    0018
                                                                START OF SEGMENT **********   11
              "B=",F12.7,"G=",F12.7,X5,"F(A,B,G)=",F12.7);                   10002300    0018
                                                                 11 IS   21 LONG, NEXT SEG     2
FORMAT FMT10(X5,"A= ",F12.7,X3,"B= ",F12.7,X3,"C= ",F12.7,                   10002400    0018
                                                                START OF SEGMENT **********   12
X5,"MIN F(A,B,C)= ",F12.7);                                                  10002500    0018
                                                                 12 IS   17 LONG, NEXT SEG     2
    FORMAT FMT11(4(F12.7,","));                                              10002600    0018
                                                                START OF SEGMENT **********   13
                                                                 13 IS    7 LONG, NEXT SEG     2
PROCEDURE TIEMPO;                                                            10002700    0018
BEGIN                                                                        10002800    0018
REAL ARRAY T[0:2];                                                           10002900    0018
                                                                START OF SEGMENT **********   14
FORMAT FMT1(///,"PROCESSOR TIME=",F9.6,/,"I O TIME=",F9.6,/,                 10003000    0001
                                                                START OF SEGMENT **********   15
            "TOTAL TIME=",F12.6,///);                                        10003100    0001
                                                                 15 IS   21 LONG, NEXT SEG    14
    T[0]←TIME(2);                                                            10003200    0001
    T[1]←TIME(3);                                                            10003300    0003
    T[2]←T[0]+T[1];                                                          10003400    0005
```

```
        T[0]←T[0]/60.;                                  0007
        T[1]←T[1]/60.;                                  0009
        T[2]←T[2]/60.;                                  0011
    WRITE(PRNT,FMT1,T[0],T[1],T[2]);                    0013
END OF TIEMPO;                                          0026
                                        14 IS   30 LONG, NEXT SEG    2
  PROCEDURE INPUT;                                      0018
    BEGIN                                               0018
        READ(CRDS,/,N,D);                               0018
        FOR I←1 STEP 1 UNTIL N DO                       0028
BEGIN                                                   0030
        READ(CRDS,/,X[I],Y[I]);                         0030
        Y[I]←Y[I]×D;                                    0039
    END;                                                0041
        READ(CRDS,/,K,DEL,RHO,DELTA);                   0044
        FOR I←1 STEP 1 UNTIL K DO                       0055
    BEGIN                                               0057
        READ(CRDS,/,PSI[I]);                            0057
    END;                                                0065
        WRITE(PRNT,FMT1,N);                             0068
        FOR I←1 STEP 1 UNTIL N DO                       0075
    BEGIN                                               0076
        WRITE(PRNT,FMT2,X[I],Y[I]);                     0076
    END;                                                0085
        WRITE(PRNT,FMT3,K,DEL,RHO,DELTA);               0087
        FOR I←1 STEP 1 UNTIL K DO                       0099
    BEGIN                                               0100
        WRITE(PRNT,FMT5,PSI[I]);                        0100
    END;                                                0108
TIEMPO;                                                 0110
        WRITE(PRNT[PAGE]);                              0111
  END OF INPUT;                                         0113
  REAL PROCEDURE EVALUATE(PHI,X,Y,N,K);                 0114
    VALUE PHI,X,Y,N,K;                                  0114
  REAL ARRAY PHI,X,Y[0];                                0114
  INTEGER K,N;                                          0114
  BEGIN                                                 0114
    REAL ARRAY FX[0:100];                               0114
                                        START OF SEGMENT **********   16
    REAL SUM,ADD,QUAN;                                  0001
    INTEGER I;                                          0001
    LABEL  THEE,THOU;                                   0001
        IF PHI[1]≤0 THEN GO TO THOU;                    0001
        IF PHI[2]≤0 THEN GO TO THOU;                    0003
        SUM←0;                                          0004
        FOR I←1 STEP 1 UNTIL N DO                       0005
    BEGIN                                               0007
        QUAN←X[I]-PHI[3];                               0007
        IF QUAN<0 THEN                                  0008
    BEGIN                                               0009
    THOU:                                               0010
        EVALUATE←100000;                                0010
        GO TO THEE;                                     0010
    END                                                 0013
    ELSE                                                0013
        FX[I]←1.0-EXP(-(X[I]-PHI[3])*PHI[2]/PHI[1]);    0013
        ADD←FX[I]-FX[I-1];                              0020
        ADD←ADD×D;                                      0022
        SUM←SUM+(Y[I]-ADD)*2/ADD;                       0023
    END;                                                0026
        EVALUATE←SUM;                                   0028
  THEE:                                                 0029
```

```
END OF EVALUATE;                                                    0030
                                                  16 IS   36 LONG, NEXT SEG    2
PROCEDURE E;                                                        0114
BEGIN                                                               0114
       FOR I←1 STEP 1 UNTIL 2 DO                                    0114
  BEGIN                                                             0115
       PHI[I]←PHI[I]+DEL;                                           0115
       SPHI←EVALUATE(PHI,X,Y,N,K);                                  0117
       IF SPHI<SS THEN SS←SPHI ELSE                                 0120
  BEGIN                                                             0122
       PHI[I]←PHI[I]-2×DEL;                                         0123
       SPHI←EVALUATE(PHI,X,Y,N,K);                                  0125
       IF SPHI<SS THEN SS←SPHI ELSE PHI[I]←PHI[I]+DEL;              0129
  END;                                                              0134
  END;                                                              0134
END OF E;                                                           0136
       WRITE(PRNT[NO]);                                             0136
  INPUT;                                                            0139
       CLOSE(CRDS,RELEASE);                                         0140
       KEEP←DEL;                                                    0142
       STOR[1]←PSI[1];                                              0142
       STOR[2]←PSI[2];                                              0144
  ITER:IF PSI[3]<0 THEN GO TO FINAL;                                0145
       PSI[1]←STOR[1];                                              0147
       PSI[2]←STOR[2];                                              0149
       DEL←KEEP;                                                    0150
       NO←0;                                                        0151
 START:SPSI←EVALUATE(PSI,X,Y,N,K);                                  0152
   ONE:SS←SPSI;                                                     0155
  COMMENT ESTABLISH BASE POINT;                                     0156
       FOR I←1 STEP 1 UNTIL K DO                                    0156
  BEGIN                                                             0158
       PHI[I]←PSI[I]                ;                               0158
  END;                                                              0159
       NO←NO+1;                                                     0161
  COMMENT MAKE EXPLORATORY MOVES;                                   0163
  E;                                                                0163
  COMMENT ARE EXPLORATORY MOVES SUCCESSFUL;                         0163
       IF SS<SPSI THEN                                              0163
  BEGIN                                                             0164
  COMMENT IF SUCCESSFUL MAKE PATTERN MOVE;                          0164
   TWO:FOR I←1 STEP 1 UNTIL 2 DO                                    0164
  BEGIN                                                             0166
       THETA←PSI[I];                                                0166
       PSI[I]←PHI[I];                                               0167
       PHI[I]←2×PHI[I]-THETA;                                       0168
  END;                                                              0171
  COMMENT END OF PATTERN MOVE,NEW POINT PHI IS THE NEW BASE POINT;  0173
       SPSI←SS;                                                     0173
       SS←SPHI←EVALUATE(PHI,X,Y,N,K);                               0174
  COMMENT MAKE EXPLORATORY MOVES;                                   0178
       NO←NO+1;                                                     0178
  E;                                                                0179
  COMMENT IF SUCCESSFUL MAKE NEW PATTERN MOVE,OTHERWISE GO TO 1 AND 0180
       TRY NEW EXPLORATORY MOVES;                                   0180
       IF SS<SPSI THEN GO TO TWO ELSE GO TO ONE;                    0180
  END                                                               0181
  ELSE                                                              0181
  COMMENT IF EXPLORATORY MOVES WERE NOT SUCCESSFUL REDUCE STEP SIZE 0181
       UNTIL THESE ARE < THAN DELTA;                                0181
       IF DEL> DELTA THEN                                           0181
  BEGIN                                                             0183
```

```
                        DEL←RHO×DEL;                                                                           10015600     0183
                        GO TO ONE;                                                                             10015700     0184
                   END;                                                                                        10015800     0185
                        WRITE(PRNT,FMT10,PHI[1],PHI[2],PHI[3],SS));                                            10015900     0185
                        WRITE(WWS,FMT11,PHI[1],PHI[2],PHI[3],SS));                                             10016000     0198
                        PSI[3]←PSI[3]-.01;                                                                     10016100     0211
                        GO TO ITER;                                                                            10016200     0213
                   FINAL:                                                                                      10016300     0215
               TIEMPO;                                                                                         10016400     0215
               END.                                                                                            10016500     0215
                                                                                                     2 IS  219 LONG, NEXT SEG     1
      EXP IS SEGMENT NUMBER 0017,PRT ADDRESS IS 0107
      LN IS SEGMENT NUMBER 0018,PRT ADDRESS IS 0106
      OUTPUT(W) IS SEGMENT NUMBER 0019,PRT ADDRESS IS 0072
      BLOCK CONTROL IS SEGMENT NUMBER 0020,PRT ADDRESS IS 0005
      INPUT(W) IS SEGMENT NUMBER 0021,PRT ADDRESS IS 0075
      X TO THE I IS SEGMENT NUMBER 0022,PRT ADDRESS IS 0110
      ALGOL WRITE   IS SEGMENT NUMBER 0023,PRT ADDRESS IS 0014
      ALGOL READ    IS SEGMENT NUMBER 0024,PRT ADDRESS IS 0015
      ALGOL SELECT  IS SEGMENT NUMBER 0025,PRT ADDRESS IS 0016
                                                                                                     1 IS    2 LONG, NEXT SEG     0
                                                                                                    26 IS   69 LONG, NEXT SEG     0
NUMBER OF ERRORS DETECTED =   0.  COMPILATION TIME =   16 SECONDS.
PRT SIZE =  76; TOTAL SEGMENT SIZE =   498 WORDS; DISK SIZE =  33 SEGS; NO. PGM. SEGS =  26
ESTIMATED CORE STORAGE REQUIREMENT =  4787 WORDS.
```

The following is a listing of a computer program used to plot the Weibull density function for any given set of parameter values.

```
GEORGIA TECH RECC B-5500 ALGOL COMPILER    SATURDAY,   6/17/67,  1:36 AM.

BEGIN                                                                        10000100    0000
                                                                   START OF SEGMENT **********     2
COMMENT  PROGRAM TO PLOT A GRAPH OF ONE OR TWO FUNCTIONS ON                  10000200    0000
         THE SAME COORDINATE SYSTEM IF BOTH FUNCTIONS ARE DEFINED            10000300    0000
         FOR A CLOSED INTERVAL FROM 0 TO 11.  THE ABSCISSA SCALE IS FROM     10000400    0000
         0 TO 11 AND THE ORDINATE SCALE IS FROM 1.5 TO -1.5.                 10000500    0000
         IF ONLY ONE FUNCTION IS TO BE PLOTTED THEN SET THE SECOND           10000600    0000
         FUNCTION EQUAL TO 2.0 (OFF THE GRAPH).                              10000700    0000
         PLACE FUNCTIONS BETWEEN COMMENTS    ;                               10000800    0000
DEFINE   FUNCTIONTOBEGRAPHED = (                                             10000900    0000
         COMMENT 1ST FUNCTION SHOULD APPEAR AFTER THIS CARD ;                10001000    0000
BB×(X-G)*(BB-1)/AA×EXP(-(X-G)*BB/AA)                                         10001100    0000
         COMMENT 1ST FUNCTION SHOULD APPEAR BEFORE THIS CARD ;               10001200    0000
         )# ;                                                                10001300    0000
DEFINE   SECFUNC = (                                                         10001400    0000
         COMMENT 2ND FUNCTION SHOULD APPEAR AFTER THIS CARD ;                10001500    0000
2                                                                            10001600    0000
         COMMENT 2ND FUNCTION SHOULD APPEAR BEFORE THIS CARD ;               10001700    0000
         )# ;                                                                10001800    0000
    INTEGER K,L;                                                             10001900    0000
INTEGER  M, I, A, C, E, GRAPH, VARIABLE, MULTIPLIER, J, COUNT ;              10002000    0000
INTEGER  XX ;                                                                10002100    0000
    INTEGER ARRAY KK[0:900];                                                 10002200    0000
INTEGER ARRAY  YY[0:2] ;                                                     10002300    0001
REAL     N, X, P, B ;                                                        10002400    0003
    REAL MIN;                                                                10002500    0003
    REAL ARRAY RR[0:700];                                                    10002600    0003
REAL ARRAY  R[0:23],  Y[0:2] ;                                               10002700    0005
REAL ARRAY  KEEPEROFX,KEEPEROFYVARIABLE,KEEPEROFYGRAPH[0:700] ;              10002800    0008
ALPHA ARRAY  POINT[0:115] ;                                                  10002900    0011
FILE IN CRD 4(2,10);                                                         10003000    0012
LABEL ENDFILE,ONE,ANFANG;                                                    10003100    0016
REAL ARRAY PHI[0:3,0:100];                                                   10003200    0016
REAL BB,AA,G,SS;                                                             10003300    0018
INTEGER JJ,NN;                                                               10003400    0018
FORMAT FMT1(/,"THE ABOVE GRAPH HAS THE FOLLOWING PARAMETERS:",               10003500    0018
                                                                   START OF SEGMENT **********     3
/,X5,"A= ",F12.7,/,X5,"B= ",F12.7,/,X5,"C= ",F12.7,/,X5,"F(A,B,C)= ",        10003600    0018
    F12.7);                                                                  10003700    0018
                                                                      3 IS   29 LONG, NEXT SEG     2
FILE OUT MCGOUT 4(2,15) ;                                                    10003800    0018
    LABEL JUMP,JUMP1;                                                        10003900    0021
LIST     MCG01(FOR C ← 0 STEP 1 UNTIL 11 DO R[C]) ;                          10004000    0021
LIST     MCG02(N,FOR E ← 0 STEP 1 UNTIL 110 DO POINT[E]) ;                   10004100    0031
LIST     MCG03(FOR E ← 0 STEP 1 UNTIL 110 DO POINT[E]) ;                     10004200    0041
FORMAT   MCG04(F5.2,111(A1)) ;                                               10004300    0050
                                                                   START OF SEGMENT **********     4
                                                                      4 IS    7 LONG, NEXT SEG     2
FORMAT   MCG05(X5,111(A1)) ;                                                 10004400    0050
                                                                   START OF SEGMENT **********     5
                                                                      5 IS    7 LONG, NEXT SEG     2
FORMAT MCG06(F7.1,11(F10.1));                                                10004500    0050
                                                                   START OF SEGMENT **********     6
                                                                      6 IS    7 LONG, NEXT SEG     2
FORMAT   MCG07("X=",F8.5," Y[VARIABLE]=",F10.5," Y[GRAPH]=",F9.5) ;          10004600    0050
                                                                   START OF SEGMENT **********     7
                                                                      7 IS   12 LONG, NEXT SEG     2
FORMAT   MCG09(3(/),"TOTAL POINTS PLOTTED = ",I3) ;                          10004700    0050
                                                                   START OF SEGMENT **********     8
```

```
B IS     11 LONG, NEXT SEG     2
        WRITE(MCGOUT[NO]);                                             0050
JJ←0;                                                                  0053
ONE: JJ←JJ+1;                                                          0054
READ(CRD,/,PHI[1,JJ],PHI[2,JJ],PHI[3,JJ],SS) [ENDFILE] ;               0055
GO TO ONE;                                                             0070
ENDFILE: NN←JJ-1;                                                      0071
CLOSE(CRD,RELEASE);                                                    0073
JJ←1;                                                                  0075
ANFANG: AA←PHI[1,JJ];                                                  0075
BB←PHI[2,JJ];                                                          0077
G←PHI[3,JJ];                                                           0079
        B ← 0 ;                                                        0081
        FOR A ← 0 STEP 1 UNTIL 11 DO                                   0082
  BEGIN  R[A] ← B ;                                                    0083
        B ← B×10 ;                                                     0084
        B ← B+5 ;                                                      0085
        B ← B/10 ;                                                     0086
  END;                                                                 0088
        MULTIPLIER ← 12 ;                                              0090
        VARIABLE ← 1 ;                                                 0091
        COUNT ← -1 ;                                                   0091
    FOR YY[GRAPH]←  60 STEP -1 UNTIL -1 DO                             0092
BEGIN   I ← 0 ;                                                        0100
        Y[GRAPH] ← YY[GRAPH]/100 ;                                     0101
        FOR XX←0 STEP 5 UNTIL 550 DO                                   0103
    BEGIN                                                              0105
             X ← XX/100;                                               0105
        IF X<G    THEN                                                 0106
    BEGIN                                                              0107
        Y[VARIABLE]←0;                                                 0107
        GO TO JUMP ;                                                   0108
    END;                                                               0109
             Y[VARIABLE] ← FUNCTIONTOBEGRAPHED ;                       0109
    JUMP:    KEEPEROFYVARIABLE[COUNT+1] ← Y[VARIABLE] ;                0118
             IF ABS(Y[GRAPH]-Y[VARIABLE])≤0.005 THEN                   0121
                  Y[VARIABLE] ← Y[GRAPH] ;                             0123
             IF Y[GRAPH]=Y[VARIABLE] THEN POINT[I] ← "*" ;             0125
             IF Y[GRAPH]=Y[VARIABLE] THEN                              0128
          BEGIN                                                        0129
             COUNT ← COUNT+1 ;                                         0129
             KEEPEROFX[COUNT] ← X ;                                    0131
             KEEPEROFYGRAPH[COUNT] ← Y[GRAPH] ;                        0132
          END ;                                                        0133
             IF Y[GRAPH] ≠ Y[VARIABLE] THEN                            0133
             BEGIN                                                     0135
                  IF Y[GRAPH]=0 THEN POINT[I] ← "-" ;                  0135
                  IF Y[GRAPH] ≠ 0 THEN POINT[I] ← " " ;                0138
             END ;                                                     0141
             I ← I+1 ;                                                 0141
    END ;                                                              0142
        IF Y[GRAPH]=0 THEN                                             0144
        FOR M←0 STEP 10 UNTIL 110 DO                                   0145
                  IF POINT[M]="-" THEN POINT[M] ← "+" ;                0148
        IF Y[GRAPH]≠0 THEN IF POINT[0]=" " THEN POINT[0] ← "1" ;       0153
        N ← ( 5×MULTIPLIER) ;                                          0157
        N ← N/100 ;                                                    0158
        IF Y[GRAPH]=N THEN                                             0159
        BEGIN                                                          0160
          IF POINT[0]="1" THEN POINT[0] ← "+" ;                        0161
          WRITE(MCGOUT[NO],MCG04,MCG02) ;                              0164
          MULTIPLIER ← MULTIPLIER-1 ;                                  0167
```

```
        END ;                                                   0168
        IF Y[GRAPH]≠N THEN                                      0168
        BEGIN                                                   0169
          IF Y[GRAPH]=-0.01 THEN WRITE(MCGOUT,MCG06,MCG01) ;    0170
          IF Y[GRAPH]≠-0.01 THEN WRITE(MCGOUT[NO],MCG05,MCG03) ; 0175
        END ;                                                   0180
        I ← 0 ;                                                 0180
        VARIABLE ← VARIABLE+1 ;                                 0180
        IF VARIABLE=2 THEN                                      0182
       BEGIN                                                    0182
        FOR XX←0 STEP 5 UNTIL 550  DO                           0183
            BEGIN                                               0185
                  X ← XX/100;                                   0185
                  Y[VARIABLE] ← SECFUNC ;                       0186
    JUMP1:        KEEPEROFYVARIABLE[COUNT+1] ← Y[VARIABLE] ;    0187
                  IF ABS(Y[GRAPH]-Y[VARIABLE])≤0.005 THEN       0190
                        Y[VARIABLE] ← Y[GRAPH] ;                0192
                  IF Y[GRAPH]=Y[VARIABLE] THEN                  0194
                BEGIN                                           0195
                        IF POINT[I] ≠ "*" THEN POINT[I] ← "x" ; 0195
                        IF POINT[I] = "*" THEN POINT[I] ← "X" ; 0198
                        COUNT ← COUNT+1 ;                       0201
                        KEEPEROFX[COUNT] ← X ;                  0202
                        KEEPEROFYGRAPH[COUNT] ← Y[GRAPH] ;      0203
                END ;                                           0205
                  IF Y[GRAPH]≠Y[VARIABLE] THEN                  0205
                        POINT[I] ← " " ;                        0206
                  I ← I+1 ;                                     0208
            END ;                                               0209
              VARIABLE ← 1 ;                                    0211
              IF Y[GRAPH] ≠ -0.01 THEN WRITE(MCGOUT,MCG05,MCG03) ; 0212
       END ;                                                    0217
END;                                                            0217
        WRITE(MCGOUT[PAGE]);                                    0220
        FOR I←0 STEP 1 UNTIL COUNT DO                           0222
    BEGIN                                                       0224
        MIN←KEEPEROFX[1];                                       0224
        K←1;                                                    0225
        FOR J←0 STEP 1 UNTIL COUNT DO                           0225
    BEGIN                                                       0227
        IF KEEPEROFX[J]<MIN THEN                                0227
    BEGIN                                                       0228
        MIN←KEEPEROFX[J];                                       0228
        K←J;                                                    0229
    END;                                                        0230
    END;                                                        0230
        KK[I]←K;                                                0232
        RR[K]←KEEPEROFX[K];                                     0233
        KEEPEROFX[K]←1000000;                                   0235
    END;                                                        0236
        FOR K←0 STEP 1 UNTIL COUNT DO                           0238
    BEGIN                                                       0241
        KEEPEROFX[K]←RR[K];                                     0241
    END;                                                        0242
        FOR J←0 STEP 1 UNTIL COUNT DO                           0244
    BEGIN                                                       0246
        L←KK[J];                                                0246
        WRITE(MCGOUT,MCG07,KEEPEROFX[L],KEEPEROFYVARIABLE[L],   0247
        KEEPEROFYGRAPH[L]);                                     0253
    END;                                                        0258
        WRITE(MCGOUT[PAGE]);                                    0260
        COUNT ← COUNT+1 ;                                       0263
```

```
                 WRITE(MCGOUT,MCG09,COUNT) ;                                                         10017100     0264
            WRITE(MCGOUT,FMT1,AA,BB,G,SS);                                                           10017200     0272
                 WRITE(MCGOUT[PAGE]);                                                                10017300     0284
            JJ←JJ+1;                                                                                 10017400     0287
            IF JJ ≤NN THEN GO TO ANFANG;                                                             10017500     0288
            END.                                                                                     10017600     0289
                                                                                         2 IS   293 LONG, NEXT SEG     1
      EXP IS SEGMENT NUMBER 0009,PRT ADDRESS IS 0111
      LN IS SEGMENT NUMBER 0010,PRT ADDRESS IS 0110
      OUTPUT(W) IS SEGMENT NUMBER 0011,PRT ADDRESS IS 0103
      BLOCK CONTROL IS SEGMENT NUMBER 0012,PRT ADDRESS IS 0005
      INPUT(W) IS SEGMENT NUMBER 0013,PRT ADDRESS IS 0107
      X TO THE I IS SEGMENT NUMBER 0014,PRT ADDRESS IS 0112
      GO TO SOLVER   IS SEGMENT NUMBER 0015,PRT ADDRESS IS 0106
      ALGOL WRITE    IS SEGMENT NUMBER 0016,PRT ADDRESS IS 0014
      ALGOL READ     IS SEGMENT NUMBER 0017,PRT ADDRESS IS 0015
      ALGOL SELECT   IS SEGMENT NUMBER 0018,PRT ADDRESS IS 0016
                                                                                         1 IS     2 LONG, NEXT SEG     0
                                                                                        19 IS    69 LONG, NEXT SEG     0
NUMBER OF ERRORS DETECTED =    0.  COMPILATION TIME =   21 SECONDS.
PRT SIZE =  78; TOTAL SEGMENT SIZE =   437 WORDS; DISK SIZE =  26 SEGS; NO. PGM. SEGS =  19
ESTIMATED CORE STORAGE REQUIREMENT =  6166 WORDS.
```

BIBLIOGRAPHY

1. Abadie, J. "Non-Linear Programming," North Holland Publishing Company, 1967.

2. Berretoni, J. N. "Practical Applications of the Weibull Distribution," Industrial Quality Control, August, 1964, pages 71-79.

3. Cochran, W. C. "The Chi-Square Test of Goodness of Fit," Ann. Math. Stat., Volume 23, pages 315-345 (1952).

4. Hess, S. W., Weaver, J. B., et al. "Nonpartisan Redistricting by Computer," Operations Research, Volume 13, pages 998-1006.

5. Hooke, Robert and Jeeves, T. A. "Direct Search Solutions to Numerical and Statistical Problems," J. Assoc. Compt. Mach., Volume 3 (1961), pages 212-229.

6. Nagel, S. A. "Simplified Bipartisan Computer Redistricting," Stanford Law Review, Volume 17, pages 863-899 (1965).

7. Meyer, P. L. "Introductory Probability and Statistics," Admission-Wesley Publishing Company, 1966.

8. Wine, R. L. "Statistics for Scientists and Engineers," Prentice-Hall, Inc., 1964.